GRC Practitioner Guide

SURECLOUD®

www.surecloud.com

19 years of expertise powers the intelligence in our industry-leading GRC platform.

YOUR BUSINESS ASSURED

Guide Contents

1. Foreword and Introduction

Most GRC teams already know what “good” should look like. The hard part is turning that picture into something people can run every day across risk, compliance, security, privacy, audit, and third-party functions.

This guide is for practitioners who sit in the middle of that challenge.

You might be a GRC manager trying to bring several programs together. You might own risk (internal or third-party), compliance, cyber audit or privacy and need to connect with the others. You may already have tooling in place, or you might still be working from spreadsheets. In all cases, your job is to turn intent into something that works in practice.

This guide does **not** aim to describe every best practice in theory. It focuses instead on what “good enough to be useful” looks like, using a small set of shared concepts that you can scale over time:

- A common set of concepts across your program
- A few simple, repeatable workflows
- A single view of work, even if the underlying data still lives in multiple systems

You can read the guide end-to-end, but most practitioners will use it as a reference when they are:

- Designing or refreshing a risk or control framework
- Preparing for a regulatory or audit cycle
- Aligning security, privacy, and third-party work to the organization’s most important services
- Planning the next step in their GRC maturity roadmap

Each chapter stands on its own, but all use the same underlying structure and language so you can join them up inside your operating model and platform.

How to Use This Guide

At the start of each chapter you will see a short prompt:

This chapter is for you if...

Use that prompt to decide whether to read it now or return to it when you are working on that part of the program.

Most chapters follow a consistent pattern:

- **A Plain-English purpose:** what this GRC practice is really for
- **Core concepts:** the minimum needed to align with other teams
- **Workflows:** steps you should be able to describe and repeat
- **Connections :** how this area connects to the rest of GRC or wider business
- **Metrics:** a small set of signals that show progress
- **Practical next steps:** actions you can take in the next few months

You do **not** need to implement everything at once. In fact, most teams begin with incomplete or inconsistent data, competing priorities, and a mix of manual and automated work. Use this guide to:

- Choose one or two domains where better structure will make a visible difference
- Align teams around shared language and the same core objects
- Decide which parts to digitize or automate first in your GRC platform

You will know the guide is working if:

- People from different teams begin using the same words for risks, controls, and issues
- It becomes easier to explain your work to senior stakeholders and auditors
- Changes in one area (for example, a new regulation or major incident) show up quickly in the others

This is intended to be a practical, living reference. As regulations shift, technologies evolve, and your own program matures, you can update the details while keeping the core model stable.

2. GRC Fundamentals: Shared Language and Building Blocks

This chapter is for you if...

Use this chapter if you:

- Have lots of GRC activity but no common way to describe it
- See that “risk”, “control”, or “issue” mean different things depending on who you ask
- Want a starting point that is realistic for your current maturity
- Need a simple model that risk, compliance, cyber, privacy, third-party, and audit teams can all reuse

This chapter is the foundation the rest of the guide assumes.

Why Fundamentals Matter

Most organisations already do some form of GRC work, even if it doesn’t feel coordinated:

- A risk register living in a spreadsheet
- Compliance descriptions embedded in policies
- Evidence held in inboxes and shared drives
- Control activities split across IT, Security, and Operations
- Vendor information scattered across procurement, IT, and local teams

The challenge is not to “invent GRC”.

The challenge is to create **shared language and repeatable building blocks** so people across the organisation can connect their work without starting from scratch each time.

A practical GRC foundation does three things well:

- Uses the same core concepts across teams
- Follows a small number of repeatable operating patterns
- Gives new work somewhere to land, even when data is incomplete

This chapter provides that foundation.

Start From the Reality You Have (Not the Ideal You Wish For)

Many teams imagine a perfect future state first:

A complete inventory of services, risks, controls, vendors, and obligations, all mapped neatly together.

Most programs do not start there.

They start from a pile — overlapping, inconsistent information spread across tools, documents, and teams.

This is normal.

And it is not a blocker.

A more realistic approach is:

- **Accept the pile:** Your starting data will be uneven and incomplete
- **Define small, stable improvements:** Even two or three shared concepts create clarity
- **Fill it gradually:** Progress matters more than completeness
- **Let the model grow through use:** Edges first, detail later

Think of this like starting a jigsaw puzzle:

- ✓ You don't complete every piece in order.
- ✓ You find the edges, group the colours, and build islands that eventually connect.
- ✓ Treat your GRC foundations the same way.

The Minimum Building Blocks You Need

To make work join up across risk, compliance, cyber, privacy, third-party, and audit, you eventually need comprehensive processes.

You do **not** need all of them on day one.

The minimum set of **shared concepts and components** are:

- **Context and scope**
  - What the organisation delivers, through which services and processes
- **Risks**
  - Scenario-based descriptions of what could happen and who or what would be affected
- **Controls**
  - Actions, behaviours, or configurations designed to keep risk within acceptable limits
- **Issues and actions**
  - Confirmed problems plus the work required to resolve them
- **Evidence**
  - Artefacts that show controls, processes, and decisions exist and operate

These components appear throughout the rest of the guide.

Later chapters introduce additional concepts — such as obligations, vendors, engagements, and processing activities — once the fundamentals are in place.

At this stage, the goal is not completeness, but **consistency**:

- Use the same definitions
- Reuse them across domains
- Build one shared model your workflows, reporting, and platform can rely on

Three Practical Starting Paths

The biggest misconception in early-stage GRC is that you must “map everything” upfront.

Instead, pick **one entry path**, keep the scope deliberately small, and complete a full cycle.

Path A: Start from your most important business area or services

This path works well if you want to align **risk and compliance** activities around business impact.

Start with 3–5 critical services or a clearly defined business area:

- Define the service or area
- List its main processes
- Identify a small number of practical risks
- Capture the key controls that matter today

This gives you a meaningful slice of your operating model that multiple teams can contribute to.

Path B: Start from your top vendors

This path is useful when third-party risk and resilience are pressing concerns.

- Identify your most critical or high-risk suppliers
- Capture what they do for you and why they matter
- Link each vendor to the services and data they support
- Note known issues or dependencies

This becomes the foundation for a usable third-party risk model.

Path C: Start from one control domain or standard

If regulatory pressure or audit readiness is the trigger:

- Select one domain (for example, information security or privacy)
- Build a simple internal control set
- Map controls to owners, scope, and evidence
- Reuse that structure across other areas over time

The golden rule

Choose a **narrow, visible, achievable scope** and run a **complete cycle end to end**.

A “cycle” means taking one defined slice of work — such as a service, vendor group, or control domain — through:

- identification
- assessment
- action
- evidence
- review

Completing one slice properly is far more valuable than sketching the entire universe.

Roles in Practice: Who Does What (and Why It Matters)

Every GRC program relies on several distinct types of contribution.

Titles, reporting lines, and organisational structures vary, but the underlying roles remain consistent.

What matters is not the label, but that **accountability is clear and workflows actually function**.

Business and service owners

These are the people who run the organisation day to day.

They:

- Own services, processes, and vendor relationships
- Make operational and delivery decisions
- Create and manage most risks in practice
- Operate the majority of controls as part of normal work

From a GRC perspective, this group owns the **activity and outcomes**.

If risks and controls are not owned here, they will not hold in reality.

GRC and risk specialists

This group includes risk, compliance, security governance, privacy, and resilience specialists.

They:

- Define methods, standards, and expectations
- Translate laws and frameworks into workable processes
- Design risk and control approaches
- Provide challenge, guidance, and consistency
- Track issues, actions, and follow-through

They own the **model**, not all the risks.

Leadership and executives

Senior leaders:

- Set objectives and risk appetite
- Decide where to invest, accept risk, or change direction
- Resolve trade-offs when priorities conflict

They rely on GRC for clarity, not detail.

Independent assurance

This includes internal audit, external auditors, and regulators.

They:

- Test whether governance, risk management, and controls work as described
- Provide independent confidence (or challenge) to leadership and boards

Their role is to **test reality**, not to design or operate controls.

Roles at a glance

Role group	What they own	Typical activities	Example job titles
Business & service owners	Services, processes, vendors, outcomes	Operate controls, manage incidents, deliver services	Service Owner, Product Owner, Head of Operations
GRC & risk specialists	Methods, frameworks, consistency	Risk assessments, control design, issue tracking	GRC Manager, Risk Manager, Privacy Officer
Leadership & executives	Direction, appetite, priorities	Approve appetite, fund initiatives, set strategy	CIO, CISO, COO
Independent assurance	Independent confidence	Audit, testing, external reviews	Internal Auditor, External Auditor

Across organisations, a few principles consistently hold:

- The business owns the activity
- GRC owns the structure and consistency
- Leadership owns direction and trade-offs
- Assurance tests the truth

The Two Loops That Run Through Everything

Almost all GRC work already follows the same basic operating pattern — **without teams necessarily realising it**.

You design how work should happen, and then you run that design repeatedly as real work arrives.

This guide describes that pattern as two nested loops: a **program loop** and an **execution loop**.

The Program Loop (Designing How Work Should Happen)

This loop shapes your frameworks, controls, and governance.

It sets the rules that day-to-day work operates within.

- **Set direction**
  - Define business objectives, key risk themes, applicable obligations, and risk appetite (the level of exposure leadership is willing to accept).
- **Design controls and processes**
  - Decide how work should be done in practice and who is responsible.
- **Implement and evidence**
  - Embed expectations into systems and workflows; define what evidence is required.
- **Review and improve**
  - Use incidents, findings, and metrics to refine the design over time.

This is where frameworks, standards, and appetite statements live.

The Execution Loop (How Work Flows Day to Day)

This loop runs whenever a new piece of work appears.

- **Intake and triage**
  - A new risk, vendor, incident, regulation, or change arises.
- **Organise work**
  - Assign owners and select the correct workflow.
- **Execute**
  - Perform the assessment, test, or approval.
- **Review and record**
  - Capture outcomes as updated risks, controls, issues, and evidence.

This is what people experience through tickets, tasks, and workflows.

Why these loops matter

Every chapter in this guide is simply these two loops applied to a specific domain.

A mature GRC program:

- Makes both loops visible
- Uses them consistently across domains
- Scales them without reinventing the model each time

Your platform should help show where work sits in each loop — not hide it.

Your First 90 Days Using These Fundamentals

If you are establishing or refreshing your GRC foundations, a realistic starting plan is:

- Pick one starting path (services, vendors, or a specific domain)
- Agree the minimum shared concepts and components you will use consistently
- Clarify who plays which role in the loops
- Run one complete execution cycle for your chosen scope
- Capture issues, actions, and evidence in the shared model
- Review what worked and refine the structure

If, after 90 days:

- People use the same terms
- New work has somewhere to land
- One small process works end to end

Then your fundamentals are in place.

You are ready to move into the domain-specific chapters that follow.

3. Risk Management Excellence

This chapter is for you if...

Use this chapter if you:

- Need a single, reusable approach to risk across operational, cyber, privacy, third-party, or enterprise domains
- Want a practical way to turn vague concerns into structured scenarios
- Need a simple lifecycle that is easy to communicate and repeat
- Want risk information that actually drives prioritisation and decision-making

This chapter builds directly on the shared language and loops introduced in Chapter 2.

What Risk Management Is Really For

Risk management is not about maintaining long lists of threats or colouring in heat maps.

Its core purpose is to help the organisation answer four simple questions:

- What could materially affect our services, customers, or strategy?
- Where should we focus first, given limited time and budget?
- What is being done about the risks that matter most?
- How do we know whether exposure is improving or worsening over time?

Risk management is also not a single technique or scoring model. Organisations use different risk assessment methodologies depending on maturity, data availability, and decision needs. The most common distinction is between qualitative and quantitative approaches.

Qualitative risk assessment

Most GRC programs start here. Qualitative approaches use defined scales (for example, low / medium / high or 1–5) to assess likelihood and impact based on:

- Expert judgement
- Past incidents and near misses
- Known weaknesses in controls
- Observable trends

This method is well suited to operational, cyber, privacy, and third-party risk, where impacts include service disruption, customer harm, regulatory consequences, and reputational damage—not just financial loss.

Quantitative risk assessment

Quantitative approaches aim to express risk in numerical terms, often using ranges or distributions to estimate potential loss or impact. In cyber and IT risk, this increasingly focuses on business outcomes such as:

- Business interruption and recovery costs
- Regulatory and legal exposure
- Customer impact and remediation effort
- Downstream operational and reputational effects

Quantitative methods can be powerful where data quality is high and decisions require financial comparison, but they also require more effort, assumptions, and specialist input.

What matters more than the method

The choice between qualitative and quantitative assessment is less important than **consistency and decision usefulness**. A practical risk program:

- Uses a method the organisation can run reliably
- Expresses impact in business terms leaders understand
- Produces comparable results across domains
- Can evolve over time as data quality and maturity improve

A strong risk practice turns insights into decisions:

- It shapes investment
- It drives control design
- It informs where automation and monitoring matter
- It connects incidents, issues, and audit findings to the bigger picture

This chapter describes a single risk lifecycle you can run across any domain using the shared concepts introduced in Chapter 2. That lifecycle works with both qualitative and quantitative approaches: scenarios, treatments, actions, and evidence remain the same—the difference lies only in how impact and likelihood are assessed and expressed.

Most organisations begin with qualitative methods and selectively introduce quantitative techniques where they add clear value, particularly for high-impact cyber and technology risks. The scoring guidance later in this chapter reflects that approach.

The Practical Risk Lifecycle

Regardless of domain, frameworks, or maturity, most risk processes follow the same underlying lifecycle. A practical version—simple enough to run consistently across the business—looks like this.

1. Identify

New or emerging risks are surfaced from incidents, near misses, audits, change activity, vendor issues, regulatory developments, technical discovery, or horizon scanning.

2. Define

Convert each item into a structured, scenario-based risk using a common template:

- **Cause** → what initiates the event
- **Event** → what actually happens
- **Impact** → who or what is affected (services, customers, finances, operations, legal, reputation)

3. Assess

Rate the scenario using shared scales for impact and likelihood. Consider:

- Existing controls
- Known vulnerabilities or issues
- Incident history and trends
- Evidence or data that supports or contradicts your assumptions

4. Treat

Decide on and document your treatment approach, then create clear, actionable plans:

- Accept
- Reduce
- Transfer or share
- Avoid

Link treatments to named actions, owners, and timelines so decisions translate into real work.

5. Monitor

Refresh scenarios, ratings, and actions regularly using:

- New evidence
- Audits and assurance activity
- Incidents and near misses
- Changes in business activity or environment

6. Govern

Aggregate risk information into views that support decision-making at different levels:

- Service-level
- Domain-level
- Enterprise-level (executive and board)

This lifecycle becomes the backbone of your risk operating model, regardless of whether risks originate in cyber, privacy, third-party, operational, or enterprise contexts.

A Note on Newer, Exposure-Led Risk Approaches

In some domains—particularly cybersecurity—newer exposure-led approaches have emerged in recent years. One example is Gartner's Continuous Threat Exposure Management (CTEM), which frames cyber risk as a repeating cycle of:

- **Scope:** Define the services, environments, or attack surfaces to examine
- **Discover:** Identify relevant assets, people, and exposures using technical tooling and specialist analysis
- **Prioritise:** Focus on the most significant risks using qualitative or quantitative methods
- **Validate:** Test whether risks are real and exploitable through exercises, simulations, or technical testing
- **Mobilise:** Act to reduce exposure through remediation, control changes, or design decisions

CTEM is intentionally cyber-focused and operational in nature. In practice, it represents a more granular execution pattern within a defined scope, particularly well suited to technical and exposure-driven risk.

The broader risk lifecycle described above is designed to accommodate these approaches, not replace them. Discovery, validation, and mobilisation activities from CTEM-style work feed naturally into scenario-based risks, treatments, issues, and metrics that can be aggregated and governed alongside other risk domains.

Used this way, newer cyber risk methodologies strengthen the overall risk engine rather than creating a parallel one. The same services, risks, controls, issues, and evidence remain visible across cyber, enterprise risk, and assurance—only the depth and tooling used at execution level changes.

Scenario-Based Risks: Clear, Useful, and Actionable

Generic labels like "cyber risk" or "regulatory risk" are too broad to drive decisions.

Scenario-based risks create clarity and shared understanding.

A scenario should describe:

- Cause: How the risk begins
- Event: The disruptive moment
- Impact: What matters for your organisation

Examples:

- *A ransomware attack encrypts our claims platform and backups, causing a three-day outage and required regulatory notifications.*
- *A critical cloud provider experiences an outage during peak trading, causing failed transactions and customer impact.*
- *A payroll provider misconfigures access, exposing employee data and triggering investigations and notifications.*

Scenario-based risk supports:

- Better prioritisation
- Better conversations with leadership
- Better linkage to controls, vendors, and evidence
- Better decision-making when choosing treatments

Every risk in your register should follow this structure for consistency.

One Unified Risk Cycle You Can Run Every Quarter

Step 1: Scope the Risk Cycle

Set boundaries so you can complete a meaningful cycle with the time you have.

Examples:

- “Customer-facing digital services”
- “Top 20 third-party dependencies”
- “Key data platforms”
- “Critical operational processes”

A narrow scope creates focus and reduces noise.

Step 2: Discover Exposure

Collect inputs from across the business:

- Recent incidents and near misses
- Open issues and audit findings
- Assessment results
- Observations from domain leads
- Project and change portfolios
- Vendor performance or outages
- Regulatory developments

The goal is not to complete a long list but to surface “what might matter most.”

Step 3: Prioritise Scenarios

Turn the raw information into structured scenarios. Then rate them with your shared scales.

Deliverables from this step:

- A small set of priority scenarios
- Clear rationale for why they matter
- Links to services, vendors, controls, and issues
- This is where the pile becomes a more coherent picture.

Step 4: Validate the Ratings

Challenge assumptions with evidence:

- Are the controls we rely on actually working?
- Do incidents contradict our ratings?
- Do we have monitoring or logs that support or weaken our view?
- Is the scenario described realistically?

This step prevents misclassification and keeps the risk model honest.

Step 5: Mobilise Treatments

For the priority risks, define explicit, actionable responses.

Each scenario should have:

- A selected treatment option (accept/reduce/transfer/avoid)
- Named actions
- Owners
- Due dates
- Expected effect on exposure
- Links to issues or problem tickets

This is where risk turns into real work.

Step 6: Review and Communicate

Share and socialise the results with:

- Business owners
- Domain leads
- Senior stakeholders
- Change and investment committees

Focus on:

- What has changed
- Why
- What is being done
- Where leadership decisions are needed

This closes the loop and prepares you for the next cycle.

Making Risk Scoring Simple and Consistent

A typical risk scoring model includes:

Impact (Low, Medium, High or 1-5))

Impact should be considered by:

- Service interruption
- Customer impact
- Financial loss
- Legal or regulatory consequences
- Reputational effect

Likelihood (4–5 levels)

Guided by:

- Incident frequency
- Exposure to threats
- Weaknesses in controls
- Domain expert judgement

Optional: Inherent vs Residual

Use only if it supports decision-making:

- **Inherent:** Exposure if relevant controls were missing or ineffective
- **Residual:** Exposure given the controls in place today

What to avoid

- Complex formulas
- Weighted matrix logic
- Scoring systems not reflected in your platform’s configuration

The purpose of scoring is not precision.

It is **comparability** and **consistent** prioritisation.

Treatments That Lead to Actual Change

A risk treatment is not a paragraph in a register—it is a real change in how the organisation works.

For each priority risk:

Accept: Exposure is understood and falls within appetite.

Reduce: Strengthen controls, redesign processes, add automation, or remove vulnerabilities.

Transfer or Share: Use contracts, insurance, or outsourcing to move or share part of the impact.

Avoid: Stop or redesign the activity causing exposure.

For every "reduce," "transfer," or "avoid" treatment, there must be:

- A named owner
- Clear actions
- Due dates
- Evidence requirements
- Expected change in risk position

These actions should appear in the shared **issues and actions** backbone from Chapter 2, so they are visible across all domains.

Making Risk Useful at Team, Domain, and Enterprise Levels

Once scenarios, assessments, and treatments exist in a shared structure, the model becomes useful for:

Teams and Process Owners

- Local scenarios for their services
- Actions they own
- Clear links to incidents and issues

Domain Leads

- Aggregated risk for cyber, privacy, TPRM, operations, or data
- Links to controls and evidence in their domain
- Inputs for assurance and testing cycles

Executive and Board Levels

- A short list of priority risks
- Clear appetite position
- Trends over time
- Actions underway
- Dependencies on investment or change

You do not need different methods for each layer—just different views built from the same model.

Signals That Your Risk Loop Is Working

Instead of tracking how many risks you have, track:

- Percentage of critical services with well-defined scenarios
- Time from identifying a high-impact scenario to agreeing treatment
- Number of repeat incidents linked to “treated” risks
- Alignment between priority risks and investment decisions
- Trend in overdue actions
- Scenario changes based on new evidence

When risk information drives decisions, influences investments, and prevents repeat issues, your model is working.

4. Compliance and the Internal Control Framework

This chapter is for you if…

Use this chapter if you:

- Are drowning in overlapping laws, standards, and customer requirements
- Have multiple control sets that say almost the same thing in different words
- Struggle to explain how obligations translate into day-to-day controls
- Want one internal view of "how we stay compliant" that everyone can use

This chapter is about understanding obligations and designing a control framework.

The regulatory change pipeline (how new or updated rules arrive and are processed) is covered separately in the Regulatory Change Management chapter.

It builds on the shared objects from GRC Fundamentals (services, risks, controls, obligations, issues, evidence) and the risk lifecycle in Risk Management Excellence.

What Compliance Work is Really For

Compliance is not about collecting certificates and answering questionnaires.

A practical compliance function should help you:

- Know which obligations actually apply to your services, entities, and data
- Design controls that keep those obligations satisfied in normal operations
- Show regulators, customers, and auditors how those controls work in practice

A good compliance function is a bridge between:

- External expectations (laws, regulations, standards, contracts)
- Internal behavior (policies, processes, and configurations)

The internal control framework is the main tool for that bridge.

Understanding Your Sources of Obligation

Most organizations face obligations from several directions at once:

- **Laws and regulations:** For example, privacy laws, sector regulations, operational resilience rules, and AI or outsourcing regimes.
- **Standards and frameworks:** ISO, NIST, SOC 2, PCI DSS, and others that shape how you design and test controls.
- **Supervisory guidance and expectations:** Regulator statements, thematic reviews, industry codes, and best-practice guidance that define "what good looks like," even when not strictly mandatory.
- **Contracts and customer commitments:** Security addenda, data protection agreements, service level commitments, audit clauses, and specific evidence or certification requirements.
- **Internal policies and risk appetite:** Choices you make about what you consider acceptable, even if the law does not demand it.

The first task is **not** to list everything.

It is to understand:

- Where obligations **cluster** (for example, around specific services, data types, or regions)
- Which ones actually **drive how you design controls**

This avoids spending time on obligations that are technically in scope but low impact in practice.

From Obligations to a Single Internal Control Framework

If you treat every law or standard separately, you end up with multiple control lists that overlap heavily and confuse everyone.

A more sustainable pattern is to design and operate **one internal control framework** that reflects how you want the organisation to behave in practice.

Create one control set

Define a consolidated list of internal controls that describe expected behaviours, processes, and configurations across the organisation.

- Keep the set as small as possible while still covering your risk and obligation profile
- Write controls in plain language that control owners can understand and operate
- Focus on controls you can realistically evidence and test

The goal is not theoretical completeness, but practical ownership and consistency.

Map obligations to that control set

For each law, regulation, standard, or contractual requirement, map its requirements to one or more internal controls.

Many organisations start this process using pre-built control frameworks rather than designing controls from scratch. Common examples include:

- **NIST CSF (including CSF 2.0)**
- **ISO/IEC 27002**
- **SOC 2 Trust Services Criteria**
- **Secure Controls Framework (SCF)** and similar consolidated libraries

These frameworks can be valuable starting points, but they come with different levels of complexity. Even “consolidated” frameworks can range from a few dozen high-level controls to many hundreds or thousands of detailed requirements.

A pragmatic approach is to:

- Start with a **smaller, more accessible framework** (for example, a tailored subset of NIST CSF)
- Select and adapt the controls that best match your services, risks, and operating model
- Map your key obligations to that reduced internal set
- Mature over time—by expanding coverage, adopting richer libraries (such as SCF), or evolving your own internal control language as your program grows

The important point is that external frameworks are inputs, not the operating model itself.

Use the internal set everywhere

Once defined, the internal control framework becomes the backbone of your GRC program.

- Policies, procedures, assessments, audits, and tooling should all reference the same internal control IDs and wording
- External laws, standards, and frameworks become views onto that internal set, not separate control lists

This way:

- When a control changes, you can immediately see which obligations, standards, and customers are affected
- When an obligation changes, you can see which internal controls may need to be updated, tested, or re-evidenced

Over time, this approach dramatically reduces duplication, improves clarity for control owners, and makes regulatory change far easier to absorb.

Managing overlap: SCF, UCF, and Real Simplification

Many organizations turn to large industry libraries such as the **Secure Controls Framework (SCF)** or **Unified Compliance Framework (UCF)** to deal with overlapping laws, standards, and customer requirements.

These libraries can be valuable reference points. They catalogue requirements across many frameworks and help identify common themes. However, they often contain **thousands of controls**, layered at different levels of abstraction. While comprehensive, this volume is daunting to own, difficult to operate, and almost impossible for control owners to engage with meaningfully.

A more pragmatic approach is to treat large libraries as inputs, not as the final internal control framework.

In practice, that means:

- Using large libraries to understand coverage and overlap
- Identifying common control requirements that appear repeatedly across your most relevant regulations and standards
- Rationalising those requirements into a much smaller internal framework that reflects how your organisation actually operates

The aim is not to track everything that exists, but to define a control set that people can run, evidence, test, and improve.

Defining a Practical Internal Control Framework

Rather than inheriting thousands of controls, many organisations define an internal framework of **a few hundred well-designed controls that**:

- Cover the core security, privacy, operational, and resilience expectations in their sector
- Are written in plain language for control owners
- Can be tested and evidenced consistently
- Map cleanly to multiple external frameworks

This internal framework then becomes the **single source of truth**, with external standards expressed as mappings or views on top.

The SureCloud Controls Framework

The **SureCloud Controls Framework** is designed around exactly this principle.

It provides a **curated, rationalised internal control framework** that takes core and extended requirements from **10+ major standards and frameworks**, including those commonly used by regulated and technology-driven organisations, such as:

- ✓ ISO/IEC 27001 and ISO/IEC 27002
- ✓ NIST Cybersecurity Framework (including CSF 2.0)
- ✓ SOC 2 Trust Services Criteria
- ✓ NCSC Cyber Assessment Framework (CAF)
- ✓ Cyber Essentials
- ✓ And other sector-specific and regulatory expectations

Rather than exposing teams to thousands of overlapping requirements, the SureCloud Controls Framework reduces these into a **manageable internal control set** that can be prioritised, owned, and operated in practice.

This allows organisations to:

- See how one internal control supports multiple obligations and standards
- Reduce duplication across assessments, audits, and evidence collection
- Prioritise controls based on risk and service criticality
- Scale coverage over time without rebuilding the framework each time a new requirement appears

Organisations can adopt the framework as-is, tailor it to their operating model, or use it as a reference point when evolving their own internal control language.

What a Good Control Looks Like

Each control in your internal framework—whether bespoke or based on the SureCloud Controls Framework—should:

- Have a clear, plain-English description of what is expected
- Be linked to one or more shared concepts:
    - Risk scenarios
    - Obligations (laws, standards, customer requirements)
    - Services, systems, and processes

Each control should also clearly identify:

- A control owner
- How it operates (manual, automated, or hybrid)
- How often it should run or be checked
- What evidence it produces and where that evidence lives

If a control cannot be described and evidenced at this level, it will be hard to test—and even harder to rely on.

Continuous Control Monitoring and Continuous Assurance

As your internal control framework matures, you can reduce manual effort and increase confidence by moving beyond periodic, manual checking toward **continuous control monitoring and assurance**.

The core idea is not to automate everything, but to establish **regular, structured signals** that show whether controls are in place and operating as intended—across services, systems, and time.

Continuous Control Monitoring in Practice

For some controls, particularly technical or configuration-based ones, conditions can be checked automatically or semi-automatically. Examples include:

- Backup jobs completing successfully and within defined thresholds
- Multi-factor authentication enabled for specific users, roles, or systems
- Critical patches applied within agreed timeframes
- Logging and monitoring enabled on defined systems

For other controls, full automation may not be realistic, but you can still monitor whether required activities have happened. For example:

- Access reviews completed by due dates
- Risk or vendor reviews signed off by the correct owners
- Approvals recorded for changes, exceptions, or policy updates
- Training completed for defined roles

In both cases, the emphasis is on **structured signals**, not ad-hoc evidence chasing.

From Monitoring to Continuous Assurance

Continuous control monitoring is a foundation for a broader concept: **continuous assurance**.

Continuous assurance is not about achieving certification once a year. It is about maintaining an ongoing view of your organisation's control posture—showing where controls are strong, where they are weakening, and where attention is needed.

In a continuous assurance model:

- Evidence is collected and refreshed as part of normal operations, not just during audits
- Test results, exceptions, and incidents feed directly back into control confidence
- Assurance is expressed as a current view of strength and weakness, not a point-in-time pass or fail
- The organisation is always close to audit-ready, rather than working toward it under pressure

This shifts the mindset from "prepare for the audit" to "operate in a way that can always be audited."

What Continuous Assurance Looks Like for Practitioners

From a practitioner's perspective, continuous assurance means:

- Control owners can see the current health of the controls they own
- GRC and security teams can identify weakening controls before they become audit findings or incidents
- Leaders can see confidence levels by service, domain, or risk theme, rather than relying on static reports
- Audits and regulatory reviews become confirmation exercises, not discovery exercises

Importantly, this does **not** require real-time monitoring everywhere. Many controls will still be evidenced periodically. The value comes from having those signals **visible, current, and connected**, rather than buried in documents and inboxes.

Designing Controls for Monitoring and Assurance

Not all controls are suitable for continuous monitoring. What matters is that controls are designed so monitoring and assurance are possible.

This means each control should clearly define:

- What "good" looks like in practice
- What signals or evidence indicate the control is operating
- How often those signals should be refreshed
- How exceptions, failures, or missed activities are handled

Detailed testing techniques and formal audit approaches are covered in the **Internal Audit Integration** chapter. The focus here is on designing controls so that monitoring, testing, and assurance can scale over time.

Scope, Traceability, and Evolution

Three recurring challenges determine whether continuous assurance is achievable in practice.

Scope

You need to know where each control applies: which entities, services, locations, and systems.

Without clear scope, you cannot answer questions such as:

- "Are we compliant in this region?"
- "Does this control apply to that business unit or service?"

Traceability

You need to see how each control links to risks, obligations, and evidence.

This allows you to answer:

- "Which controls support this regulation or customer requirement?"
- "What evidence shows this control is operating today?"

Traceability is what turns monitoring signals into confidence.

Evolution

Controls change as the business, technology, and regulatory landscape evolve.

You must be able to update:

- Control definitions
- Ownership and scope
- Mappings to obligations and risks
- Evidence expectations

without breaking reporting, assurance, or audit history.

A GRC platform should support this by keeping controls, links, evidence, and changes in a single, connected model—rather than scattered across documents and spreadsheets.

Interfaces with Other Chapters

The internal control framework is one of the main threads running through this guide:

Risk Management Excellence: Controls are the levers you pull to keep priority scenarios within appetite.

Regulatory Change Management: New or changed obligations are assessed and mapped back into the internal control set.

Cyber Risk and Resilience, TPRM, and Privacy: Domain-specific controls (for example, encryption, vendor oversight, or RoPA accuracy) are expressed within the same framework.

Internal Audit and Assurance: Tests and reviews check whether controls are designed and operating as described; findings feed back into control definitions, ownership, and confidence levels.

By keeping one internal control framework at the centre—and designing it for monitoring and assurance—you avoid each domain building its own isolated control universe and make it far easier to scale automation, assurance, and trust over time.

5. Regulatory Change as an Operating Discipline

This chapter is for you if...

Use this chapter if you:

- Feel blindsided by new or changing regulations
- See the same “we missed this requirement” story repeat across regions or entities
- Struggle to show who noticed a change, who assessed it, and what decision was taken
- Need to manage regulatory change across EMEA and the U.S. while keeping risk, privacy, and third-party teams aligned

This chapter is not about tooling or automation.

It is about building a repeatable operating discipline for absorbing regulatory and contractual change as your organisation matures.

The interpretation of obligations and the design of controls are covered in the Compliance and Internal Control Framework chapter. This chapter focuses on how change flows through your program over time.

What Regulatory Change Discipline Is Really For

Regulatory change discipline exists to answer four questions, consistently and defensibly:

- What has changed in the laws, regulations, guidance, or contracts that apply to us?
- Does it matter for our services, entities, data, vendors, or operating model?
- What decisions did we take, and what needed to change as a result?
- How can we show our reasoning, actions, and outcomes later?

A mature approach reduces surprises. It turns:

> *“We found out in an audit letter.”*

into:

> *“We identified this early, assessed its impact, made a decision, and can show our rationale and follow-through.”*

This is not a standalone capability. It is an application of the **same program and execution loops** described earlier, applied to regulatory signals instead of incidents or risks.

Where Regulatory Change Actually Comes From

In practice, regulatory and compliance expectations enter organisations through several channels at once:

1. **Laws and regulations**
   New or amended acts, directives, rules, and technical standards.
2. **Regulator expectations and guidance**
   Supervisory statements, thematic reviews, speeches, and informal guidance that reshape what "good enough" looks like.
3. **Customer and contractual requirements**
   Security addenda, resilience clauses, audit rights, and data protection terms—often where emerging expectations surface first.
4. **Internal policy and appetite changes**
   Shifts in what leadership is willing to accept, even when external rules remain unchanged.

A mature program treats all of these as **change signals**, even when they do not come in the form of formal legislation.

Global Reality: Change Rarely Arrives Neatly

For global organisations, regulatory change rarely lands uniformly.

- In **EMEA**, expectations around privacy, resilience, outsourcing, and AI are tightening, with more emphasis on governance, testing, and vendor oversight.
- In the **U.S.**, federal rules may move more slowly, but state laws, sector guidance, and customer contracts continue to raise the bar.

The result is:

- Asynchronous change across regions
- Greater reliance on third parties and cloud providers
- Increased pressure on privacy, TPRM, and resilience teams

A regulatory change discipline must reflect this reality, rather than assuming a single-country or single-framework view.

A Simple Regulatory Change Process (Process, Not Tooling)

You do not need a complex system to manage regulatory change. What matters is a **clear, repeatable process** that fits your operating model.

A pragmatic pipeline looks like this:

1. Scan and capture

Identify potential changes through legal, compliance, regulators, industry bodies, and key customers.

Record each change in a shared register so it does not depend on individual memory.

2. Triage and route

Decide quickly whether the change is:

- Not relevant
- Monitor only
- Potentially material

Assign ownership and route material items to the appropriate domain leads.

3. Assess impact

For material changes, assess:

- Which services, entities, regions, data, or vendors are affected
- Which existing obligations, controls, policies, or contracts are touched
- The likely impact on risk, effort, and timelines

Record assumptions and decisions, not just conclusions.

4. Decide and plan

Agree whether the response is:

- No action (with rationale)
- Minor adjustment
- Material change to controls, contracts, or operating model

Capture actions, owners, and dates in the same issues and actions backbone used elsewhere.

5. Implement and evidence

Update controls, policies, contracts, training, or mappings as required.

Store evidence in predictable locations.

6. Review and close

Confirm actions were completed and had the intended effect.

Close the item with a short record of what was done and why.

This process is the same whether the trigger is a regulation, guidance note, customer contract, or internal policy shift.

Change Discipline vs Obligation Design

The **Compliance and Internal Control Framework** chapter covers how you design and document obligations and controls.

Regulatory change discipline is about **when and how that design is revisited**.

To keep the operating model clear:

- Use this process to handle signals, assessments, and decisions
- Use the control framework as the authoritative record of the current state

If change stops at "we noted the law changed," the organisation will still be surprised later.

Interaction with Third Parties

Many regulatory changes affect third parties as much as internal teams.

A mature process explicitly asks:

- Does this affect vendor selection, tiering, or oversight?
- Does it require contract or DPA changes?
- Does it introduce new expectations for critical or high-risk engagements?

The key is not to create a separate TPRM process, but to **flag when vendor governance must be involved**.

Metrics That Show Maturity

Useful indicators focus on discipline and follow-through, not volume:

- Time from identifying a material change to a documented assessment
- Percentage of material changes with clear ownership and decisions
- Number of late surprises identified by audits, regulators, or customers
- Percentage of changes where third-party and privacy impacts were explicitly considered

These metrics show whether the organisation is **learning to absorb change**, not just reacting.

Practical Next Steps

To strengthen regulatory change discipline:

- Agree a simple way to capture and triage change signals
- Define what “material” means for your organisation
- Pilot the process on a small number of real changes
- Ensure outcomes flow into controls, policies, and issues—not just slides

When this discipline is in place, regulatory change becomes another input into your operating model, rather than a recurring fire drill.

6. Cyber Risk and Resilience

This chapter is for you if…

Use this chapter if you:

- Have many security tools and controls, but struggle to explain your real exposure
- Run technical tests and incident drills, but the lessons don't consistently land in GRC
- Need to connect cyber work to services, continuity, and third-party risk
- Want to move from "secure on paper" to "able to absorb and recover from disruption"

This chapter treats cyber risk as **operational resilience in practice**, not as a separate technical universe. It applies the scenario-based risk model and risk lifecycle from **Risk Management Excellence** to the cyber domain.

What Cyber Risk and Resilience are Really For

Cybersecurity is not only about preventing attacks. It is about:

- Keeping important services running
- Limiting the impact when incidents occur
- Recovering quickly enough that customers, regulators, and partners can tolerate the disruption

A practical cyber and resilience program helps you:

- Understand which services and processes would hurt most if disrupted
- Anticipate likely attack paths and failure modes
- Prepare the organization to detect, contain, and recover from incidents
- Learn from events and exercises so the same weaknesses do not repeat

From a GRC perspective, this means treating cyber risk as **business-impact scenarios**, not just a stream of vulnerabilities, alerts, or dashboards.

From Technical Issues to Business Scenarios

Start by framing cyber risk in terms that business leaders recognize. For example:

- A ransomware attack encrypts a core platform and its backups, causing a multi-day outage and regulatory notifications.
- Credentials stolen from a third-party support engineer are used to access production, leading to customer data exfiltration.
- A misconfigured cloud storage bucket exposes sensitive reports to the internet for several weeks.

For each scenario, identify:

- The services and entities affected
- The data involved
- Downstream impacts on customers, operations, revenue, and regulatory obligations

This connects cyber directly to the **scenario structure** in the risk chapter (cause → event → impact) and uses the shared objects from Fundamentals (services, risks, controls, vendors, issues, evidence).

Threat-led Identification and Continuous Exposure Thinking

Most security teams already run a range of **threat-led activities**, for example:

- Vulnerability scanning and penetration testing
- Red and purple team exercises
- Threat hunting and log analysis
- Cloud configuration and posture assessments

Instead of treating these as isolated technical tasks, use them as **inputs to the same risk lifecycle** you defined in Chapter 3.

A common pattern is:

- **Scope**
  - » Focus on specific services, environments, or attack surfaces: external web apps, critical cloud workloads, identity systems, etc.
- **Discover**
  - » Identify exposures, misconfigurations, vulnerabilities, weak or missing controls.
- **Prioritize**
  - » Link findings to the scenarios and services that matter most; avoid long, unranked lists.
- **Validate**
  - » Confirm what an attacker could realistically do and test the effectiveness of key controls.
- **Mobilize**
  - » Turn validated exposures into tracked issues and remediation plans, with owners and target dates.

This is simply the **generic risk loop** from Chapter 3 applied to cyber exposures, not a separate framework. The critical part for GRC is that **exposures and test findings land in the same issues and actions register** used by the rest of the program, linked to risks and controls.

Readiness: exercises, training, and communication plans

You cannot prevent every incident, so **readiness** is as important as prevention. That includes:

- **Tabletop exercises**
  - » Scenario-based discussions walking through how an event would unfold.
  - » Include technical teams and business stakeholders (legal, communications, operations, HR).
- **Technical simulations and cyber ranges**
  - » Hands-on exercises that test detection and response under realistic conditions.
- **Playbooks and runbooks**
  - » Agreed steps and roles for responding to common incident types (ransomware, vendor breach, lost device, cloud misconfiguration).
- **Communication plans**
  - » Clear guidelines for talking to leadership, regulators, customers, and staff during incidents.
  - » Alignment with legal and PR on who communicates what, when, and via which channels.

From a GRC perspective, the **outputs** of these activities matter as much as the exercises themselves:

- Updated risks and scenarios
- Refined controls, playbooks, and responsibilities
- Logged issues and actions with owners and due dates

If the only record of an exercise is a slide deck, the learning will fade before the next event.

Making Exercises Count in Your GRC System

To ensure exercises and lessons learned actually improve your posture:

- Record each exercise as an event in your incident or assurance log
- Capture, at minimum:
  - » Scenario tested
  - » Services, entities, and vendors involved
  - » What worked well, what failed
  - » Recommendations and decisions taken
- Log resulting actions in the **shared issues and actions register**, and link them to:
- Relevant cyber and enterprise risks
- Controls and playbooks that need updating
- Owners in both technical and business teams

This way, time spent on readiness translates into **visible changes** in your risk and control landscape.

Linking Cyber to Continuity and Third-party Risk

Cyber incidents rarely stay within IT. They often overlap with:

- **Business continuity and disaster recovery**
  - » Data center or cloud region outages
  - » Loss of identity platforms or communication channels
- **Third-party risk events**
  - » Outages or breaches at SaaS providers, MSPs, and infrastructure vendors
  - » Compromised third-party accounts with access to your systems

In your GRC model:

- Map critical services to both internal and external dependencies (systems, vendors, data centers, regions).
- Link cyber and continuity risks to the **same service and vendor objects** used by TPRM and enterprise risk.
- Make sure continuity plans and vendor incident processes reference the **same roles and communication patterns** as your cyber incident playbooks.

This lets you answer questions like:

- “Which top risks involve this cloud provider?”
- “Which services are at risk if this identity platform fails?”

and avoids cyber, continuity, and TPRM running in parallel universes.

Metrics That Matter for Cyber and Resilience

Useful metrics focus on **readiness and learning**, not just “number of alerts” or “tools deployed”. Examples:

- Percentage of critical services with defined cyber and continuity scenarios and named owners
- Time to detect, contain, and recover from major incidents affecting those services
- Number of significant cyber incidents or near misses per quarter, and how many led to changes in controls or playbooks
- Coverage of tabletop and technical exercises across critical services and vendors in the last 12–18 months
- Percentage of high-priority cyber and resilience actions completed on time

These metrics should be shareable in both **security and enterprise risk** forums.

Practical next steps

To strengthen cyber risk and resilience within your GRC program:

- Choose one or two critical services and write clear cyber-impact scenarios using the risk model from Chapter 3.
- Map those services to their key systems and vendors, and confirm they appear in your vendor and engagement inventory.
- Run (or review) at least one tabletop exercise about a realistic scenario (for example, a major vendor outage or ransomware against a core system), and log outputs as issues and actions.
- Agree a small set of cyber resilience metrics to add to regular risk reporting, focusing on detection, response, and recovery rather than tool counts.

Handled this way, cyber risk and resilience become part of how you manage c**ore services**, not a separate specialist function reporting from the sidelines.

7. Third-Party Risk Management (TPRM)

This chapter is for you if…

Use this chapter if you:

- Know you rely heavily on vendors and platforms but can't see your real exposure
- Have a list of suppliers but no clear view of which ones you depend on most
- Find that vendor risk assessments are one-off events rather than part of a lifecycle
- Struggle to keep up when vendors change scope, process new data, or add new services over time

This chapter treats third-party risk as a core part of operational resilience, not a side spreadsheet. It applies the shared objects from **GRC Fundamentals** (services, vendors/engagements, risks, controls, issues, evidence) and the **risk lifecycle** from **Risk Management Excellence** to your supply chain.

What Third-party Risk Management is Really For

Third-party risk management is not about stopping people from using vendors. It is about helping the organization:

- Understand how critical services depend on external providers
- Make informed decisions about who to trust with what, and on what terms
- Detect when vendor risk changes and respond before it becomes an incident
- Show regulators, customers, and boards that supply chain risks are managed, not guessed

In practice, that means treating third-party risk as **scenario-based risk**:

- What could happen **through** a vendor
- How it would affect your **services, customers, and obligations**
- What you are doing about it, and how you will know if exposure changes

In 2026, very few services are purely "internal." Most sit on top of layers of external technology and service providers.

A More Realistic View of Your Supply Chain

It is no longer enough to say "we have many suppliers." The real shift is dependence:

- **Horizontal platforms**
  - » Cloud providers, identity platforms, payment gateways, communication tools, core SaaS.
  - » Outages or breaches can affect many services at once.
- **Vertical and sector-specific providers**
  - » Industry-specific software, data providers, logistics partners, and specialized outsourcers.
  - » Failures can damage your ability to operate in particular markets or lines of business.
- **Physical and technology mix**
  - » Real-world providers (couriers, facilities, data centers) intertwined with digital services.

This mix creates **cascading risk**: a single issue at one provider can ripple across multiple services, regions, and customers. TPRM has to make those dependencies visible, not just record names and contracts.

Vendor vs Engagement: Capturing How Scope Really Changes

A common failure mode in TPRM is treating each vendor as a single, static risk. In reality, what matters is the **engagement**:

- **Vendor**
  - » The legal entity or group (for example, "GlobalCloud Inc.").
  - » Has attributes like financial health, jurisdiction, and broad security posture.
- **Engagement**
  - » The specific service or set of services you buy from that vendor, with its own:
  - » Purpose and business owner
  - » Data types and volumes processed
  - » Regions and hosting locations
  - » Access types (for example, admin access, VPN, API, on-site staff)
  - » Interfaces with your systems and processes

Over time, engagements change:

- A vendor that started by providing a simple SaaS tool starts processing more sensitive data.
- A supplier takes on additional roles, such as consulting, support, or on-site work.
- A new integration gives them direct access to production systems.

If you treat vendor risk as a one-time "tick box" at onboarding, you will miss this scope creep.

A practical TPRM model:

- Stores **vendors and engagements separately**
- Tracks changes in each engagement's data, access, and service criticality
- Ties assessments and risk levels to the **engagement**, not just the vendor name

In data-model terms, "vendor" and "engagement" are just specialized forms of the shared **service and dependency objects** introduced in Chapter 2.

A Simple TPRM Lifecycle

You do not need a complex method to get value from TPRM. A clear lifecycle is enough:

1. **Identify and map**
   - » Build and maintain an inventory of **vendors and engagements**.
   - » Link each engagement to the **services, entities, and data** it supports.

This is where you turn an unstructured "pile" of suppliers into a more reliable picture of your supply chain.

2. **Tier and scope**
   - » Classify engagements by:
     - Criticality to operations and customers
     - Data sensitivity and volume
     - Connectivity and access level
   - » Use these tiers to decide how **deep and frequent** assessments should be.

Tiering is your main control for keeping effort proportionate and aligned to inherent risk.

3. **Internal Review and decide**
   - » Run due diligence appropriate to the tier (questionnaires, document review, certifications, independent reports).
   - » Consider both security/privacy and financial/operational resilience.
   - » Decide whether to:
     - Proceed
     - Proceed with conditions
     - Seek alternatives

Here you are using the **risk assessment and treatment** steps from Chapter 3, but at the engagement level.

4. **Contract and onboard**
   - » Reflect risk and control expectations in contracts and data protection terms.
   - » Align internal teams on how the engagement will work in practice (access, escalation paths, reporting).

Contracting and onboarding should translate risk decisions into **controls and obligations** you can later test and evidence.

5. **Monitor and review**
   - » Track incidents, changes in scope, new data types, and changes in the vendor’s posture.
   - » Reassess on a regular cadence appropriate to the tier, and whenever something material changes.

This is the **monitor** step in the risk loop: watching for new information that should change your view of exposure.

6. **Offboard and transition**
   - » Plan how you will exit: data return and deletion, access removal, and service migration.
   - » Capture lessons learned and feed them back into criteria and templates.

Each step should produce data that lives in your GRC system (engagements, risks, controls, issues, evidence), not in scattered documents.

Capturing Change in the Relationship

The hardest part of TPRM is not the first assessment; it is staying aligned as reality changes. To handle this:

- Treat **change events** as part of your TPRM workflow, such as:
  - » New data categories or volumes
  - » New regions or hosting changes
  - » New integrations or access types
  - » New services from the same vendor, or mergers and acquisitions
- Require **engagement owners** to log these changes and trigger quick reassessments where needed.
- Connect **regulatory change items** to TPRM, so new rules about outsourcing, resilience, or data transfers can prompt updates to assessments and contracts.

This is where the vendor vs engagement split becomes critical. You might:

- Accept one engagement with a vendor
- Reject, delay, or reshape another

based on the different risk profiles.

In the shared risk lifecycle terms, change events are just **new inputs** to the **identify → assess → treat** steps for that engagement.

Interfaces With Other Chapters

Third-party risk sits at the intersection of several parts of this guide:

- **Risk Management Excellence:** Many top enterprise risks have third-party components; engagements should be linked to scenarios and treatments.
- **Compliance and Regulatory Change Management:** Outsourcing, resilience, and privacy rules often flow through vendors; TPRM must reflect changes in those expectations.
- **Cyber Risk and Resilience:** Vendor outages, breaches, and access issues are core cyber scenarios; incident and continuity plans should reference key engagements.
- **Data Privacy and Protection:** Vendor and engagement records should show who is processing what personal data, under which legal bases and contracts.
- **GRC Strategy and Maturity:** Use automation and AI to remove friction in TPRM (for example, document extraction and triage), while keeping humans accountable for decisions to accept or change vendor relationships.

Handled this way, TPRM is not an isolated process; it is one of the main ways risk, compliance, cyber, and privacy intersect around **services and dependencies**.

Metrics That Matter for Third-Party Risk

Useful TPRM metrics focus on **critical exposures and follow-through**, for example:

- Percentage of Tier 1 and Tier 2 engagements with current assessments and contracts aligned to actual scope
- Number of significant vendor-related incidents or outages per quarter, and how many led to changes in controls or contracts
- Time from identifying a material change in an engagement (new data, new region, new integration) to updated risk assessment and decision
- Coverage of critical services with mapped key vendors and engagements
- Percentage of high-priority TPRM actions completed on time

These metrics should connect directly to **resilience and risk reporting**, not sit in a separate supplier dashboard.

Practical Next Steps

To strengthen third-party risk management using this model:

- Build or refine a **vendor and engagement inventory** for a subset of critical services.
- Tier engagements based on criticality, data, and access; agree what each tier means for assessment and review effort.
- For your top few vendors, check whether current contracts and assessments match **how you actually use them today**.
- Integrate vendor and engagement data with **incident, regulatory change, cyber, and privacy** processes so changes and events flow through consistently.

Handled this way, third-party risk stops being a separate, “poor relation” process and becomes part of how you understand and protect your core services.

8. Enterprise Risk

This chapter is for you if...

Use this chapter if you:

- Have strong domain-level risk activity (cyber, third-party, privacy, resilience) but no joined-up enterprise view
- Struggle to explain “our top risks” in a way leadership recognises and acts on
- Need to connect board-level risk appetite to what is actually happening across services, vendors, and change programmes
- Want to use GRC information to steer strategy and investment, not just populate reports

This chapter focuses on **enterprise-level risk sense-making**: how information from across the organisation is pulled together into a small number of meaningful conversations for senior leaders and the board.

It assumes the shared concepts from **GRC Fundamentals** (services, risks, controls, issues, evidence) and the scenario-based lifecycle from **Risk Management Excellence**.

What Enterprise Risk Is Really For

Enterprise risk management is not about maintaining a long, static list of threats.

Its purpose is to help senior leaders and the board:

- Understand the small number of risks that could materially change the organisation’s direction
- See how those risks connect to critical services, external dependencies, and strategic choices
- Decide where to accept, reduce, or reshape exposure
- Check that major investments, programmes, and controls align with those decisions

Enterprise risk does **not** replace domain risk management. It sits above it.

Put simply:

- **Enterprise risk** is the lens leadership uses to understand what matters most
- **Domain and service-level risk** is where those exposures are identified, assessed, and treated in practice

The same scenario structure and lifecycle apply at both levels. The difference is **scope and audience**, not method.

From Domain Views to an Enterprise Narrative

Most organisations start with risk information organised by domain:

- Cyber and information security
- Third-party and supplier dependencies
- Privacy and data protection
- Regulatory and compliance exposure
- Resilience and service continuity

Each domain can be mature in its own right. The challenge is that leadership experiences risk **across the portfolio**, not in silos.

To move from domain views to an enterprise narrative:

Use scenarios as the common language

Ask each domain to express its most important risks as scenarios using the **same cause → event → impact** structure.

This removes specialist jargon and allows comparison across very different risk types.

Anchor risks to services and dependencies

Ensure each scenario is linked to the services, entities, regions, and key third parties it affects.

This keeps enterprise risk grounded in how the organisation actually operates, not abstract categories.

Look for themes and clusters

Group scenarios that point to the same underlying exposure, such as:

- Dependence on a small number of platforms or providers
- Concentration of regulatory scrutiny in specific regions or products
- Complex data flows across vendors and jurisdictions

Select a small number of top risks

Typically 8–15, depending on size and complexity.

Each top risk should be explainable in plain language and traceable back to the underlying domain scenarios that inform it.

This is where enterprise risk adds value: turning many inputs into a **manageable set of strategic conversations**.

Appetite, Alignment, and Real Decisions

Risk appetite only becomes meaningful at enterprise level if it influences real decisions.

To make that happen:

- Translate high-level appetite statements into **clear expectations**, such as:
  - Acceptable outage durations
  - Tolerance for data loss or regulatory exposure
  - Willingness to rely on single providers or platforms
- Ensure domain practices align:
  - Impact scales used in cyber, privacy, and third-party risk should roll up cleanly into enterprise impact definitions
  - "High / medium / low" should mean the same thing when discussed at board level
- Use enterprise risks as a lens on:
  - Major investments and transformation programmes
  - Outsourcing and vendor strategy
  - Market entry, product launches, and acquisitions

If risk appetite lives only in a policy or slide deck, and budgets and roadmaps tell a different story, the enterprise risk view is not yet working.

Using Risk Data to Shape the Enterprise View

Enterprise risk should be:

- **Fed by** detailed work happening in domains and services
- **Feeding back into** how priorities, funding, and attention are set

Key inputs include:

- **Incidents and events**
  Patterns in major incidents, near misses, and exercises that reveal gaps between plans and reality.
- **Issues and actions**
  Where high-priority remediation clusters or repeatedly slips.
- **Control and assurance results**
  Common findings, recurring weaknesses, or controls that fail across multiple areas.
- **Third-party and data exposure**
  Concentration of critical services on a small number of vendors or high-risk processing arrangements.

The value here is not the volume of data, but the ability to **see patterns by theme and service**, rather than by domain or team.

A Simple Enterprise Risk Rhythm

Enterprise risk does not need a complex cadence. A pragmatic rhythm might include:

- **Quarterly or biannual enterprise risk review**
  Refresh top risks, discuss trends, and confirm alignment with reality.
- **Regular domain and service reviews**
  Domains review their scenarios, incidents, and issues using the same lifecycle, escalating material changes.
- **Annual strategy and planning alignment**
  Use enterprise risks as an input into budgeting, roadmaps, and major programmes.
- **Ongoing intake of events and change**
  Feed major incidents, regulatory developments, and strategic shifts into the enterprise view as they arise.

Consistency and traceability matter more than precision in scoring.

Interfaces With Other Chapters

Enterprise risk acts as the **integration layer** across the guide:

- **Risk Management Excellence:** Provides the scenario structure and treatment discipline used everywhere.
- **Cyber, TPRM, and Privacy**: Supply many of the scenarios that shape enterprise risks.
- **Compliance and Regulatory Change:** Influence enterprise themes, appetite discussions, and priority setting.
- **GRC Strategy and Maturity**: Helps determine how sophisticated the enterprise view needs to be at each stage.

Enterprise risk does not replace these domains. It helps leadership see how they connect.

Metrics That Matter at Enterprise Level

At enterprise level, focus on whether the view is **trusted and used**, not just produced:

- Coverage of critical services by at least one enterprise risk
- Frequency and quality of leadership and board risk discussions
- Evidence that major investments and remediation programmes link directly to named enterprise risks
- Reduction in repeated, high-impact incidents aligned to top risk themes
- Alignment between enterprise risks and external disclosures or regulatory communications

If leaders use the enterprise risk view to make better decisions—and practitioners can see their work reflected in that view—this chapter has done its job.

9. Internal Audit Integration

This chapter is for you if...

Use this chapter if you:

- Feel like internal audit operates on a parallel track, with its own lists and language
- See the same findings repeat across audits, assessments, and incidents
- Struggle to turn audit reports into clear, trackable improvements
- Want assurance and audit readiness to feel like part of the GRC engine, not an annual scramble

This chapter focuses on how assurance and audit connect to the **shared GRC objects** (services, entities, risks, controls, vendors, issues, evidence) and to the **risk and control lifecycles** described in the rest of the guide.

What Internal Audit is Really For

Internal audit is not there to run a second compliance team. Its primary job is to provide independent assurance to the board and senior leadership that:

- Governance, risk management, and controls are designed sensibly
- They operate as described, over time
- The most important risks and obligations are receiving appropriate attention

For practitioners, that means:

- Audit is one of the most valuable feedback loops you have
- Findings should feed directly into your issues, actions, and control improvement work
- Assurance topics should align with your **risk scenarios, top risks, and key controls**, not surprise you

In the language of the earlier chapters: internal audit tests how well your **program loop** (design) and **execution loop** (day-to-day work) are actually functioning.

Keeping Assurance and Design Clearly Separated

The **Compliance and Internal Control Framework** chapter covers how you:

- Interpret obligations
- Design and document internal controls
- Decide who owns what

Internal audit and second-line assurance answer a different question:

> *"Are these controls, processes, and behaviors actually working as intended?"*

To keep roles clear:

- **Design and ownership** sit with first and second line
- **Testing and independent verification** sit with internal audit (third line) and, in some cases, second-line assurance teams
- **Issue management and remediation** are shared, but tracked in one place

Across this guide, assurance and audit readiness are treated as a **single, continuous activity**, not as two unrelated processes.

A Shared Issues and Actions Backbone

To avoid the “we fix it for audit only” cycle, you should have:

- **One issues and actions register** that captures:
  - Audit findings (internal and external)
  - Assurance test results
  - Incident postmortems
  - Regulatory inspection outcomes
  - Self-identified issues
- Each issue linked to:
  - A **risk scenario** and/or **top risk**
  - One or more **controls and processes**
  - The relevant **service, entity, and vendor** (where applicable)
- Simple status tracking:
  - Severity or priority
  - Owner and due date
  - Progress and verification of closure

If your audit findings live in one system, risk issues in another, and incident actions in a third, you will repeat work and lose the bigger picture. The shared register is where the **execution loop** for issues and remediation plays out, regardless of where a finding originated.

Planning Assurance Based on Risk

Internal audit and second-line assurance should focus where risk is highest and where controls matter most.

A practical approach:

- **Start with your top risks and key services**
  - Use the enterprise and domain risk views to identify the scenarios that matter most.
  - Identify the controls and processes that really shape exposure for those scenarios.
- **Build an assurance plan that covers:**
  - Risk themes (for example, data protection, third-party resilience, access management)
  - Critical controls and services
  - Regulatory priorities and past findings
- **Spread coverage over a multi-year horizon:**
  - Not everything needs to be reviewed every year.
  - Use risk and incident data to decide what to review more or less often.
- **Coordinate with second-line teams:**
  - Avoid duplication between routine testing, thematic reviews, and formal audits.
  - Share work where possible (for example, reusing testing results if they are robust).

The goal is a plan that leadership can recognize as aligned to what they care about, not just a list of topics. It should look like an extension of your **risk and control program**, not a separate calendar.

Making Audits Easier to Run and Easier to Consume

You can make audit and assurance work smoother by:

- **Using the same object model**:
  - » Scope audits by service, entity, vendor, and control, not just by department.
  - » Reference the internal control framework directly in scoping and testing.
- **Standardizing test and evidence expectations**:
  - » For each control, agree acceptable evidence, frequency, and test types.
  - » Store or reference evidence in known locations, not scattered files.
- **Keeping reporting focused on**:
  - » What was tested and why (linked to risks and controls)
  - » What was found and how severe it is
  - » What needs to change, by when, and who owns it
- **Feeding results into the issues and actions register automatically**, not retyping them into spreadsheets.

This reduces friction for both auditors and control owners, and makes it easier to see patterns over time, especially when combined with **continuous control monitoring** where it makes sense.

Closing the Loop: From Findings to Learning

Assurance and audit add value only if findings lead to real improvements. To close the loop:

- Treat **root-cause analysis** as a standard step for significant findings and repeated issues
- Look for patterns across domains:
  - » Are the same themes (for example, weak change control, unclear ownership, poor data quality) appearing in different audits and incidents?
- Use that insight to:
  - » Refine control definitions and responsibilities
  - » Adjust training and guidance
  - » Update risk scenarios and appetite discussions
  - » Inform the GRC strategy and maturity roadmap

Enterprise and operational risk teams should see audit and assurance outputs as a **primary input**, not just a compliance tick.

Interfaces With Other Chapters

Internal audit and assurance connect to many parts of this guide:

- **Risk Management Excellence and Enterprise Risk:**
  - » Top risks should shape the audit plan.
  - » Audit results should refine risk ratings and treatments.
- **Compliance and Internal Control Framework**:
  - » Testing focuses on whether controls are designed and operating as described.
- **Cyber, TPRM, and Privacy:**
  - » Domain-specific audits feed back into shared risks, controls, and vendor or processing records.
- **Regulatory Change Management**:
  - » Regulatory inspections and thematic reviews are a form of assurance; outcomes should flow into the same register and framework.

Seen this way, internal audit is another way of exercising the same s**hared objects and lifecycles**, not a separate discipline.

Metrics That Matter For Audit Integration

Useful metrics show whether assurance work is targeted and effective, for example:

- Percentage of top enterprise risks that have had relevant assurance work in the last 2–3 years
- Percentage of audit and assurance findings logged in the central issues register
- On-time completion rate for high-priority remediation actions
- Number of repeated findings by theme across multiple audits or years
- Time from audit report issuance to final closure of critical findings

These metrics can be shared with audit committees and leadership to demonstrate that assurance is part of a continuous improvement loop, not an isolated activity.

Practical Next Steps

To better integrate internal audit into your GRC program:

- Create or confirm a **single issues and actions register** and agree that all significant findings will be recorded there
- Align audit scoping with your **internal control framework and top risks**, using the same IDs and definitions
- Pilot this integrated approach on one or two upcoming audits, making sure findings and actions flow into the wider GRC data model
- Use results from those pilots to refine how often you meet with internal audit and how you align plans, reports, and metrics

Handled this way, internal audit becomes a powerful extension of your GRC program, strengthening confidence in what is working and shining a clear light on where you need to improve next.

10. Data Privacy and Protection

This chapter is for you if…

Use this chapter if you:

- Struggle to track who processes what personal data, where it flows, and under which legal bases
- Need a clearer connection between privacy, security, TPRM, and regulatory change
- Have RoPAs, DPIAs, and assessments scattered across documents, spreadsheets, and teams
- Want a practical way to show regulators and customers that your data protection program is structured and trustworthy

This chapter applies the **shared GRC objects** (processing activities, services, data, vendors/engagements, risks, controls, issues, evidence) introduced in Chapter 2, and the **risk lifecycle** from Chapter 3, to the world of privacy.

It treats privacy as a **business process** at the intersection of legal, technical, and third-party dependencies—not as an isolated compliance exercise.

What Data Privacy and Protection Are Really For

A strong privacy program helps the organization:

- Understand where personal data lives and how it moves
- Ensure those processing activities follow legal bases and obligations
- Identify and mitigate risks to individuals (and by extension, to the organization)
- Make good decisions about vendors, systems, and new uses of data
- Respond confidently to incidents, rights requests, and regulator questions

In practice, privacy depends on **accurate, shared information** about:

- Services and processes
- Vendors and engagements
- Data categories and volumes
- Systems, integrations, and access
- Controls, evidence, and issues

If any one team tracks this alone, gaps appear immediately.

Privacy as Part of the Shared GRC Model

In earlier chapters, several core objects were introduced:

- Services and processes
- Risks and scenarios
- Controls and evidence
- Vendors and engagements
- Issues and actions

Privacy adds two more that fit naturally into that model:

1. **Processing activities**
2. **Data categories and attributes**

All privacy work—RoPAs, DPIAs, LIAs, reviews, consent, and cross-border assessments—should anchor to these objects.

This makes privacy work **traceable**, reusable, and compatible with TPRM, cyber risk, and regulatory change.

Building a Usable Record of Processing Activities (RoPA)

A RoPA isn't valuable because it exists—it's valuable because it is accurate and used.

A practical RoPA records, for each processing activity:

- **Purpose of processing**
- **Service or business process supported**
- **Data categories and subjects involved**
- **Systems and vendors used**
- **Regions where processing occurs**
- **Legal basis**
- **Relevant controls and evidence**
- **Risks and associated DPIAs**

Instead of treating the RoPA as a large spreadsheet, use it as a **data object** that lives in your GRC or privacy platform, connected to:

- Vendors (via engagements)
- Security and TPRM assessments
- Data flows
- Issues and incidents
- Controls
- Regulatory change items

This makes the RoPA the backbone of your privacy program, not an administrative burden.

Data Mapping As The Foundation

You cannot do privacy well without understanding **data flows**.

Data mapping should be:

- Simple enough to maintain
- Detailed enough to support DPIAs, incident investigations, and vendor due diligence
- Linked directly to processing activities and engagements

Even if visual diagrams are produced, the **structured data** behind them should include:

- Where data enters (forms, imports, integrations)
- Where it is stored
- Where it is sent (vendors, systems, reports)
- Where it leaves the ecosystem
- Access types and roles involved

This feeds directly into cyber risk (attack surface), TPRM (vendor scope), and resilience (dependencies).

DPIAs as Structured Risk Assessments

A DPIA is not a separate risk process. It is the **generic risk lifecycle** applied to privacy-specific risks to individuals.

For each DPIA:

- **Identify:** What changes? What data? What processing? What could go wrong?
- **Define:** Describe risks as **scenarios** (cause → event → impact on individuals).
- **Assess:** Impact and likelihood using privacy-consistent scales.
- **Treat:** Controls, design changes, vendor terms, data minimization, pseudonymization.
- **Monitor:** Trigger reassessment for material changes in scope, data, vendors, or regions.
- **Govern:** Record outcomes, legal basis, and decisions; make them discoverable.

When DPIAs feed into the same **issues and actions register** used by cyber, TPRM, and audit, privacy becomes naturally embedded in the broader GRC program.

Legal Bases and Consent Management

Privacy obligations differ by jurisdiction, but most share a few core ideas:

- **Define the legal basis** for each processing activity
- **Explain the rationale** for that basis
- **Show the evidence** (consent logs, contracts, policy references)
- **Track changes** in purpose, data category, or jurisdiction

Your GRC or privacy platform should make it easy to:

- See legal bases by service, data category, vendor, or region
- Identify where consent or contract updates are required
- Link legal bases to controls and evidence

This ensures that changes in product, marketing, or customer experience naturally trigger privacy reviews—not after-the-fact corrections.

Third Parties and International Transfers

The privacy, cyber, and TPRM chapters intersect heavily when personal data leaves your environment.

You should be able to answer:

- Which engagements process personal data?
- Which data categories and subjects are involved?
- In which regions is the data stored or accessed?
- What transfer mechanism applies (SCCs, adequacy, BCRs)?
- What controls and evidence demonstrate compliance?

In the shared model:

- Vendors are **vendors**
- Their services are **engagements**
- Processing activities link to these engagements
- Cross-border rules become attributes of those engagements

This avoids duplicating work between privacy, legal, TPRM, and cyber.

Incident Handling and Breach Response

A privacy incident is a **service-impacting scenario** with additional obligations.

Use the same incident and root-cause processes described in Cyber and Enterprise Risk, but capture privacy-specific details:

- What data was affected?
- Which subjects?
- How sensitive is the data?
- Which jurisdictions?
- Which regulators require notification?
- What evidence supports your timeline and decisions?

Ensure privacy incidents flow into:

- The **central issues and actions register**
- Risk assessments (if they reveal missing or ineffective controls)
- Vendor and engagement reviews (if a third party was involved)
- Regulatory change monitoring (if the event reveals new guidance or expectations)

This prevents privacy incidents from becoming isolated investigations.

Interfaces With Other Chapters

Privacy touches almost every part of the GRC ecosystem:

- **Risk Management Excellence:** DPIAs follow the same lifecycle as other risk processes.
- **Cyber Risk and Resilience:** Most privacy risks are triggered by cyber failures; most cyber incidents have privacy impacts.
- **Third-Party Risk Management:** Vendor and engagement records feed directly into privacy assessments and cross-border decisions.
- **Compliance and Regulatory Change:** New privacy regulations and guidance flow through the same change pipeline.
- **Internal Audit Integration:** Privacy controls, processes, and decisions are subject to thematic audits and evidence testing.

Handled this way, privacy becomes a **shared language and shared model**, not a siloed documentation exercise.

Metrics that Matter for Privacy

Track metrics that reflect **accuracy, responsiveness, and maturity**, for example:

- Percentage of processing activities with complete, current RoPA entries
- Percentage of high-risk processing activities with DPIAs completed and reviewed
- Time from identifying a material change to updating RoPA, DPIA, and controls
- Number of privacy-relevant incidents, and how many triggered changes in controls or vendor terms
- Coverage of cross-border transfers and evidence of appropriate mechanisms
- Percentage of privacy-related actions completed on time

These metrics work well in both privacy steering groups and enterprise-risk reporting.

Practical Next Steps

To strengthen data privacy and protection:

- Build or refine your **processing activity inventory** using the shared GRC objects
- Map personal data categories, systems, and vendors to those activities
- Ensure high-risk activities have DPIAs aligned to the scenario-based risk model
- Link privacy controls to your internal control framework and evidence standards
- Integrate privacy checks into change, vendor onboarding, and project approval flows
- Make sure issues and actions flow into the same central register used by cyber, TPRM, audit, and risk

Handled this way, privacy becomes part of how you operate—not a compliance afterthought.

11. GRC Strategy and Maturity

This chapter is for you if…

Use this chapter if you:

- Need a clear way to describe "where we are" and "where we're going" in GRC
- Are under pressure to show progress beyond one-off projects and tool rollouts
- Want to prioritize investments instead of trying to improve everything at once
- Need a maturity model that matches how programs actually grow, not a neat theory

This chapter gives you a practical maturity view you can use with leaders, auditors, and vendors, grounded in the **shared objects** and **loops** from earlier chapters.

What a GRC Maturity Model is Really For

A maturity model is not a scorecard for marketing or an excuse to add complexity.

It should help you:

- Describe your current state in language stakeholders recognize
- Agree the next sensible step, not an ideal end state you cannot reach
- Make trade-offs clear: where you will standardize, automate, or invest next
- Keep tooling, process, and people changes aligned over time
- Tie improvements back to the **program loop** (design, implement, improve) and the **execution loop** (intake, execute, record)

The model below is designed to match what we see in real organizations, not a perfect linear journey. Some teams start in the middle. Others are advanced in one domain and basic in another.

The GRC Maturity Stages (High-level View)

You can think of GRC maturity in seven broad stages:

1. Manual and disjointed
2. Standardized and documented
3. Selective automation
4. Continuous assurance
5. Defined GRC
6. Strategy and resilience
7. Led by intelligence

You do not need to use these labels verbatim in your organization, but the patterns they describe are useful when planning.

It's More Like a Jigsaw Than A Staircase

Most organizations don't move neatly through A → B → C → D → E. It is more like completing a **jigsaw puzzle**:

- You start by finding the edges: the shared objects, roles, and basic lifecycles
- Then you build **floating islands** where the pressure is highest (for example, cyber or TPRM)
- Over time, those islands connect as services, risks, controls, vendors, and evidence are brought into the same picture

The maturity model needs to allow for that variation. It is normal for some domains to be at Stage 2 while others are at Stage 4 or 5.

Stage 1: Manual and disjointed

Typical signs:

- Risks, controls, issues, and vendors are managed in separate spreadsheets and inboxes
- Each domain (cyber, compliance, audit, privacy, TPRM) has its own way of working
- Reporting is reactive and labor-intensive; data is rarely trusted or reused

Your focus here:

- Establish the **common objects and language** from GRC Fundamentals
- Introduce the idea of **program and execution loops**, even if still manual
- Consolidate at least one key list (for example, services, risks, or vendors) into a shared view
- Begin capturing **issues and actions** in one place

At this stage, success is simply "one version of the basics" instead of many.

Stage 2: Standardized and documented

Typical signs:

- Core processes (risk assessments, policy approvals, vendor onboarding) are documented
- A basic internal control framework exists, even if it's still rough
- There is some alignment on risk scales and definitions

Your focus here:

- Tighten and rationalize the **internal control framework** so it reflects how you actually operate
- Standardize a small set of **lifecycles** (for example, risk, TPRM, incidents, regulatory change) using the shared model
- Agree simple, scenario-based risk descriptions and consistent scales
- Start moving high-friction workflows into a GRC platform to avoid duplication

The goal is to move from "every team does it differently" to "we do it the same way, even if still manual."

Stage 3: Selective automation

Typical signs:

- Key workflows are in a GRC platform, but many processes remain manual
- Some evidence and assessments are captured in structured forms
- Reporting is easier but still requires manual clean-up and interpretation

Your focus here:

- Use automation and AI where it removes the most friction in the **execution loop** (document intake, mapping, reminders, routing), while keeping humans in control of judgments in the **program loop**
- Strengthen the **issues and actions register** as the single source of truth for remediation across risk, compliance, cyber, TPRM, privacy, and audit
- Make sure data structures in the platform mirror the model in this guide:
  - » Services and processes
  - » Entities and regions
  - » Risks and scenarios
  - » Obligations and controls
  - » Vendors and engagements
  - » Issues, actions, and evidence

At this stage, the priority is to remove repetitive admin, not to automate decisions.

Stage 4: Continuous assurance

Typical signs:

- Control testing and assurance are planned with a multi-year view
- There is a visible link between **top risks, key controls,** and **assurance coverage**
- Some technical signals are used to monitor control health between audits

Your focus here:

- Tighten integration between internal audit, second-line assurance, and operational testing, using the **shared issues and actions backbone**
- Expand continuous control monitoring concepts where they make sense, especially for technical and configuration-based controls
- Make sure findings from all forms of assurance feed into the same data model and are traceable to:
  - » Service(s) affected
  - » Risk scenarios and top risks
  - » Controls and obligations
  - » Vendors and engagements where relevant

Continuous assurance at this stage means “steady, predictable feedback loops,” not real-time automation everywhere.

Stage 5: Defined GRC

Typical signs:

- GRC is recognised as a shared capability, not just a collection of projects
- Roles and responsibilities across the business, GRC specialists, leadership, and assurance are clear and widely understood
- The GRC platform is used consistently across key domains

Your focus here:

- Align GRC strategy with **business objectives and risk appetite**, not just regulatory demands
- Use the maturity model to plan improvements **by domain** (for example, moving TPRM from Stage 2 to 3 while enterprise risk moves from 3 to 4)
- Start measuring outcomes in terms of **risk, resilience, and trust,** not just activity counts (number of assessments, policies, or findings)
- Use the shared objects and lifecycles as the default way to design new processes and integrate new tools

At this stage, the conversation shifts from “what are we doing?” to “is it making a difference?”

Stage 6: Strategy and resilience

Typical signs:

- Top risks, resilience themes, and investment decisions are clearly linked
- Third-party, cyber, privacy, and continuity work is coordinated around **critical services**
- GRC is part of strategic planning, M&A, and major transformation programs

Your focus here:

- Use your GRC data (scenarios, incidents, issues, metrics) to shape strategic choices:
  - Where to expand
  - Where to partner or outsource
  - Where to simplify or exit
- Embed risk and resilience thinking in **product, change, and procurement** processes from the start
- Develop playbooks for major shocks that draw on the full program:
  - Risk and enterprise top risks
  - Continuity and crisis management
  - Security and privacy
  - Third parties and supply chain
  - Communications and stakeholders

Here, GRC becomes a way to test and refine strategy, not just report on it.

Stage 7: Led by intelligence

Typical signs:

- Data from across GRC and security is used to **forecast and simulate** risk under different conditions
- Targeted AI assistance helps summarize, prioritize, and route work, while humans own judgments and approvals
- Leadership uses GRC insights as part of regular performance and strategy discussions

Your focus here:

- Strengthen data quality and shared models so analytics and AI have something solid to work on:
  - Clean, linked objects (services, risks, controls, vendors, evidence)
  - Clear ownership and lifecycle states
- Introduce **agentic patterns** carefully for administrative and triage tasks with clear guardrails, approvals, and auditability
- Keep governance, transparency, and human oversight central as you scale intelligent assistance

Not every organization needs to reach this stage in every domain. The goal is to know where this approach would genuinely add value and where simpler methods are enough.

It's Not Linear, and That's Okay

Few programs move cleanly from Stage 1 to Stage 7. In practice:

- Cyber may be at Stage 4 while TPRM is at Stage 2
- Privacy might be strong on documentation (Stage 2–3) but weak on integration with enterprise risk
- Some regions or entities may have more advanced practices than others

You can use the maturity model:

- Per domain (cyber, TPRM, privacy, enterprise risk, internal audit)
- Per region or business unit
- For the overall GRC program

What matters is that you know **where you are, what the next step looks like**, and how it plays through the program and execution loops—not that you score highly in every box.

Interfaces With Other Chapters

This chapter pulls together themes from the rest of the guide:

- Use **Risk Management Excellence** and **Enterprise and Operational Risk** to shape top-risk and appetite discussions at each stage.
- Use **Compliance and Internal Control Framework** and **Regulatory Change Management** to define what "good enough" looks like for obligations and controls as you mature.
- Use **Cyber Risk and Resilience**, **Third-Party Risk Management**, and **Data Privacy and Protection** to understand how domain capabilities move through the stages at different speeds.
- Use **Internal Audit Integration** to align assurance work, themes, and findings with maturity goals rather than treating audits as separate events.

Handled this way, the maturity model becomes a **navigation tool** for the whole guide.

Practical Next Steps

To use this maturity model in your own program:

- Run a simple self-assessment across a few domains using the stage descriptions
- Validate the assessment with a small group of stakeholders from the first, second, and third line
- Agree one or two improvements per domain for the next 12–18 months, rather than trying to move everything at once
- Make sure each improvement clearly links back to:
  - Shared objects (services, risks, controls, vendors, evidence)
  - The program and execution loops
  - Outcomes you can explain in terms of risk, resilience, and trust
- Use your GRC platform to track maturity-related actions and metrics alongside day-to-day work

Handled this way, GRC maturity becomes a tool for focus and alignment, not just a slide in a board pack.

12. Glossary of GRC Terms

Core GRC and Risk

GRC: Governance, Risk, and Compliance; how an organization is directed, how uncertainty is managed, and how obligations are met.

Risk: The effect of uncertainty on objectives; what could happen and how it would affect services, customers, and strategy.

Risk appetite: The amount and type of risk leadership is willing to take in pursuing objectives, usually expressed in high-level statements.

Risk thresholds/tolerances: Specific limits that show when a risk has gone too far and needs action.

Risk scenario: A concrete description of a risk using cause → event → impact.

Scenario-based risk: An approach that describes risks as specific events and impacts rather than broad categories.

Risk register: A structured list of risk scenarios, ratings, treatments, and links to services and owners.

Risk treatment: The decision taken for a risk: accept, reduce, transfer/share, or avoid.

Accept (risk treatment): Choose to live with the risk within appetite, with no major changes.

Reduce (risk treatment): Strengthen or add controls, or change processes and designs, to lower risk.

Transfer/share (risk treatment): Use contracts, insurance, or partnerships so some impact is borne by others.

Avoid (risk treatment): Stop, delay, or redesign the activity so the risk no longer exists in its current form.

Inherent risk: The level of risk before controls are considered.

Residual risk: The level of risk after existing controls and treatments are taken into account.

Control (internal control): A behavior, process, configuration, or rule designed to keep risk within appetite or support compliance.

Control owner: The role responsible for making sure a control is defined, implemented, and kept up to date.

Manual control: A control performed by people rather than technology alone.

Automated control: A control performed by systems or tools without manual steps.

Hybrid control: A control that combines manual and automated elements.

Issues and actions register: The shared list of issues and follow-up actions across audits, incidents, assessments, and regulatory work.

Program loop: The higher-level cycle where you set direction, design controls, implement, and improve based on feedback.

First line: Business and operations teams that own services, processes, vendors, and most risks and controls in practice.

Second line: Risk, compliance, security, privacy, and TPRM teams that set standards, methods, and provide challenge and support.

Third line: Internal audit, providing independent assurance that governance, risk management, and controls work as intended.

GRC platform: A technology platform that supports GRC processes and objects in one model.

Enterprise and Operational Risk

Enterprise risk management: Organization-wide risk management focused on the few risks that could materially change direction or strategy.

Top risks: A small set of higher-impact risks that leadership agrees to track, discuss, and use to steer decisions.

Risk themes: Clusters of scenarios that point to the same underlying exposure.

Near miss: An event that could have been a significant incident but was caught or limited before major impact.

Incident: An event that disrupts normal operations or causes a meaningful deviation from expected performance or security.

Material change: A change significant enough that it should trigger re-assessment, decision, and updates to controls, contracts, or records.

Enterprise risk review: A periodic leadership forum that refreshes top risks, discusses trends, and confirms alignment with reality.

Enterprise risk operating rhythm: The overall cadence of enterprise reviews, domain reviews, planning cycles, and ongoing intake of incidents and regulatory changes.

Cyber Risk and Resilience

Cyber risk: Risk of loss, disruption, or harm arising from attacks, misuse, or failures involving information systems and data.

Operational resilience: The ability to keep important services running and recover quickly when disruptions occur.

Cyber resilience: The aspect of resilience focused on cyber incidents, attacks, and technical failures.

Ransomware: Malware that encrypts data or systems and demands payment or other concessions to restore access.

Attack path: The route an attacker could take through systems, identities, and integrations to achieve an impact.

Vulnerability scanning: Automated checking of systems for known weaknesses that attackers could exploit.

Penetration testing: Authorized attempts to exploit vulnerabilities to understand what an attacker could realistically achieve.

Red teaming: Simulated attacker exercises focused on testing detection and response under realistic, adversarial conditions.

Purple teaming: Collaborative exercises where defenders and simulated attackers work together to test and strengthen detection and response.

Threat hunting: Proactive search through data and systems for signs of compromise or suspicious behavior.

Attack surface: The systems, services, and interfaces that are exposed to potential attackers.

Cyber incident: An incident with a primary cause in security failures such as compromise, exfiltration, ransomware, or misconfiguration.

Tabletop exercise: A discussion-based simulation of a scenario that walks through how an incident would unfold and how teams would respond.

Playbook: A predefined set of steps and roles for responding to a specific incident type or scenario.

Compliance, Regulations, and Frameworks

Obligation: A specific requirement from laws, regulations, standards, contracts, or internal policies.

Standards and frameworks: Structured sets of requirements or controls such as ISO, NIST, SOC 2, PCI DSS, and others.

Supervisory guidance: Regulator statements, expectations, and industry codes that shape what "good enough" looks like.

Internal policy: A high-level rule that sets expected behavior or direction inside the organization.

Internal control framework: A single, rationalized list of internal controls that map to risks and obligations and are reused across domains.

Regulatory change management: The process for spotting, assessing, deciding on, and implementing regulatory and contractual changes over time.

Regulatory change pipeline: The defined steps (scan, triage, assess, decide, implement, review) that each change passes through.

SCF (Secure Controls Framework): A large, industry control library often used as input when designing a condensed internal control framework.

UCF (Unified Compliance Framework): A commercial framework that harmonizes many external requirements into a unified control view.

PCI DSS: Payment Card Industry Data Security Standard; sets requirements for handling cardholder data and related environments.

SOX: Sarbanes-Oxley Act; drives requirements around financial reporting controls, access, and auditability.

SOC 2: An attestation framework that reports on controls related to security, availability, processing integrity, confidentiality, and privacy.

Regulatory inspection: A review or examination by a regulator or authority to check how well obligations are being met.

AI governance requirement: An obligation or expectation related to how AI is designed, used, monitored, and controlled.

Privacy and Data Protection

Privacy and data protection: The discipline focused on how personal data is collected, used, shared, stored, and protected.

Personal data: Information that relates to an identified or identifiable individual.

Data subject: The individual to whom personal data relates.

Data subject rights: Rights individuals have over their data, including access, correction, deletion, and objection.

Record of Processing Activities (RoPA): A structured inventory of processing activities, linked to services, purposes, data subjects, systems, vendors, and legal bases.

Processing activity: A distinct use of personal data for defined purposes within specific services, systems, and vendors.

Legal basis/legal bases: The legal reason for processing personal data such as contract, legitimate interest, or consent.

Data minimization: Collecting and retaining only the data needed for defined purposes and no more.

DPIA (Data Protection Impact Assessment): A structured assessment of privacy risks and mitigations for high-risk processing.

TIA (Transfer Impact Assessment): An assessment of privacy and legal risks when transferring personal data across borders, especially from stricter to less strict regimes.

Privacy incident: An incident where personal data is exposed, misused, or processed contrary to obligations or expectations.

Cross-border transfer: Movement of personal data between jurisdictions or regions with different legal regimes.

Consent mechanism: The way individuals are asked for, record, and manage consent for specific processing activities.

Sector-specific privacy obligations: Privacy rules that intersect with sector controls, such as PCI DSS or SOX, influencing access, logging, retention, and segregation.

Third-Party Risk and Vendor Management

Third-party risk: Risk created when external providers deliver services, handle data, or have access to systems on your behalf.

Third-Party Risk Management (TPRM): The lifecycle for identifying, assessing, monitoring, and treating risk in vendor and supply-chain relationships.

Vendor: The legal entity or group you buy services or products from.

Engagement: The specific service or set of services you consume from a vendor, with defined data, access, and criticality.

Vendor vs engagement: The distinction between the overall legal entity and each particular service relationship, which can have different risks.

Tiering (TPRM): Classifying engagements based on criticality, data sensitivity, access level, and resilience impact.

Due diligence: Pre-engagement checks such as questionnaires, document reviews, certifications, and independent reports.

Contract and onboarding: The stage where risk and control expectations are written into contracts and internal teams align on how the engagement will work.

Change event (TPRM): A change in data, access, region, integration, or service scope that should trigger an update to the engagement record and risk assessment.

Vendor-related incident: An incident where a vendor’s failure, outage, breach, or misconfiguration materially affects your services or data.

Horizontal platform: A widely used, cross-industry platform such as a cloud provider, identity platform, or communication tool.

Vertical/sector-specific provider: A provider focused on a particular industry, domain, or niche service.

Supply chain dependence: Reliance on layers of external technology and services such that a single provider issue can cascade across services and regions.

Business Continuity, Resilience, and Crisis

Continuity plan: A documented approach for how a service continues or recovers under disruption.

Resilience themes: Cross-cutting exposure patterns that shape resilience work, such as platform dependence or vendor outages.

Major incident: A high-impact incident affecting critical services, customers, or regulatory obligations.

Internal Audit and Assurance

Internal audit: An independent function that provides assurance to the board and leadership on governance, risk management, and controls.

Assurance: Evidence-based confidence that controls and processes are designed and operating effectively.

Second-line assurance: Testing and review activities performed by second-line teams separate from internal audit.

Audit finding: An observation from an audit or assurance review that shows a control gap, weakness, or deviation.

Assurance plan: A forward-looking plan of what controls, services, and themes will be tested over a period.

Audit scoping: The process of deciding which services, entities, vendors, and controls an audit will cover.

Shared issues and actions register: The central place where audit findings, incidents, regulatory outcomes, and self-identified issues are tracked.

Continuous assurance: An operating model where testing and signals about control health occur regularly, not just once a year.

Audit readiness: The state of having clear control definitions, mappings, and evidence so audits can be run and answered efficiently.

Automation, AI, and Exposure Management

AI assistance: Using AI to summarize, extract, prioritize, and route information, while humans still own judgments and decisions.

Mapping (controls and obligations): Linking obligations, risks, controls, and evidence in a structured, reusable way.

Continuous control monitoring: Using technical signals and structured checks to monitor whether controls are in place and operating between audits.

Exposure: A weakness, misconfiguration, or condition that could be exploited or that increases risk if left untreated.

Agentic patterns: Designs where AI can trigger and coordinate limited actions across tools within strict guardrails and oversight.

Data, Systems, and Architecture

Entities and regions: Legal entities, jurisdictions, and business units in which the organization operates.

Cloud provider: An external provider of infrastructure, platforms, or software services hosted in the cloud.

Identity platform: A platform that manages authentication, authorization, and user access for systems and services.

SaaS (Software as a Service): Software delivered over the internet and managed by a third-party provider.

Integration: A technical connection between systems that allows data exchange or process automation.

Production environment: Live systems and data that underpin real customer and business operations.

Attack surface (technical): The externally and internally exposed systems, interfaces, and configurations that attackers could target.

Metrics, Reporting, Committees, and Maturity

Risk report: A structured update on risk scenarios, treatments, incidents, and trends for a given audience.

Enterprise risk report: A risk summary tailored for senior leadership and the board, focusing on top risks, appetite, and trends.

GRC committee: A governance group that coordinates GRC priorities, reviews risks and metrics, and aligns domains.

GRC maturity model: A staged view of how GRC capabilities evolve from manual and disjointed to intelligence-led; used as a reference to plan practical next steps.

13. About SureCloud

SureCloud is a Governance, Risk, and Compliance (GRC) and cybersecurity solutions provider that helps organizations move from reactive reporting to **risk-led decision-making**.

Our cloud platform brings together risk, compliance, security, privacy, third-party, and audit workflows into a single, connected data model. Instead of running separate spreadsheets and siloed tools, teams work from one shared view of **services, risks, controls, vendors, issues, and evidence**—the same shared objects used throughout this guide.

This unified approach supports both the **program loop** (design, implement, improve) and the **execution loop** (intake, execute, record), enabling organizations to operate more coherently across domains.

What Practitioners use SureCloud For:

Organizations use SureCloud to:

- Map risks, controls, and obligations to the services that matter most
- Rationalize overlapping frameworks into a single internal control library
- Orchestrate assessments, issues, and remediation across the first, second, and third line
- Monitor critical third-party engagements and track incidents, findings, and changes over time
- Prepare for audits and regulatory reviews with clear ownership and scalable evidence management

The platform is designed to reduce friction in day-to-day work while improving visibility, consistency, and confidence in the decisions teams make.

Solutions That Scale With Your Program

SureCloud supports organizations at every stage of maturity:

SURECLOUD® | FOUNDATIONS

For growing teams, Foundations provides a streamlined way to establish a modern GRC program quickly. It includes opinionated templates, sensible defaults, and prescriptive workflows aligned to common standards.

Perfect for teams moving from s**preadsheets and email** to a structured model with shared objects and repeatable lifecycles.

SURECLOUD® | ENTERPRISE

Larger or more mature organizations can tailor and extend the platform to:

- Fit their operating model and governance structures
- Integrate with security tooling, IT systems, and business applications
- Support federated teams, regional variations, and complex workflows
- Link enterprise risk, cyber, privacy, audit, and TPRM in a single view

This flexibility supports teams moving into later maturity stages—**continuous assurance, resilience**, and **intelligence-led operations**.

Our Focus Across All Deployments

Across all programs—large or small—our mission is consistent:

- Make risk, compliance, and resilience data usable in day-to-day decisions
- Reduce manual friction without removing human judgment
- Help teams move from ad-hoc, spreadsheet-driven processes to connected, outcome-driven GRC
- Provide a practical path that aligns with the maturity model in this guide, rather than forcing organizations into unrealistic end-states

This reflects our promise: **Your Business Assured**.

For more information on how SureCloud can assist your organization,
visit us online at **www.surecloud.com**, or email **sales@surecloud.com**

SureCloud Summary

SureCloud Ltd. has two decades of experience as a leading provider of Governance, Risk, and Compliance (GRC) solutions since its founding in 2006. Headquartered in the UK, with offices in the US, SureCloud supports a global portfolio of organisations with its holistic and intelligent GRC platform. Whether addressing cyber risk, data privacy, third parties, or compliance demands, SureCloud has a proven record of empowering organisations to continuously identify, manage and automate their risk and regulatory alignment.